

## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

### LISTING OF CLAIMS

1. (Original) A method of specifying a stateful web service within a procedural programming environment, the method comprising:  
  
    providing a source code representation of at least a portion of web service logic, the logic including one or more methods;  
  
    identifying one of said one or more methods to be exposed as part of the stateful web service; and  
  
    specifying one or more declarative annotations to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified method.
2. (Original) The method of claim 1, wherein the one or more declarative annotations are specified within the source code representation.
3. (Original) The method of claim 2, wherein the one or more declarative annotations are specified within a comment field preceding the identified method.
4. (Original) The method of claim 1, wherein the one or more declarative annotations are specified outside of the source code representation and associated with the identified method by the compiler.

5. (Original) The method of claim 1, wherein the one or more declarative annotations indicate to the compiler whether the identified method is at least one of a start method, a continue method, and a finish method, wherein the start method applies to the start of a stateful conversation between a client and the web service, the continue method applies to the continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to the completion of an ongoing stateful conversation between a client and the web service.

6. (Original) The method of claim 5, wherein when a method declared to be a start method is invoked at run-time, a new instance of a conversation is created, and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

7. (Original) The method of claim 5, wherein when a method declared to be a continue method or a finish method is invoked at run-time, a unique identifier provided by the client is obtained and used to access a corresponding instance of a conversation.

8. (Original) The method of claim 7, wherein when a finish method is invoked at run-time, the corresponding instance of the conversation is destroyed after processing by the web service logic.

9. (Original) The method of claim 1, wherein the one or more declarative annotations indicate to the compiler whether the identified method is buffered, wherein if the identified method is buffered the compiler instantiates one or more queues to temporarily store one or more requests for the identified method.

10. (Original) The method of claim 1, wherein the one or more declarative annotations are manually specified by a developer.

11. (Original) The method of claim 1, wherein the one or more declarative annotations are automatically specified by an integrated development environment based upon input provided by a developer.

12. (Original) The method of claim 11, wherein said input includes graphical manipulation of the identified method by the developer via the integrated development environment.

13. (Original) The method of claim 1, wherein the one or more declarative annotations cause the compiler to generate a proxy object designed to facilitate interaction by the web service with one of an external web service or client.

14. (Original) The method of claim 13, wherein the one or more declarative annotations further cause the compiler to route asynchronous responses from the external web service to code specified by a developer of the web service.

15. (Original) The method of claim 13, wherein the one or more declarative annotations further cause the compiler to generate a unique identifier to identify a specific conversational instance of the external service.

16. (Previously Presented) In a procedural programming environment, a method of generating a stateful web service, the method comprising:

reading on one or more computing devices a segment of procedural source code

representing at least a portion of the web service;

parsing on one or more computing devices the segment of source code to identify the

presence of one or more declarative annotations identifying an associated

method within the segment as being stateful;

generating on one or more computing devices one or more object codes defining one

or more publicly accessible service components based at least in part upon the

source code;

generating on one or more computing devices meta-data based at least in part upon

the one or more declarative annotations;

associating on one or more computing devices the meta-data with the one or more

object codes.

17. (Previously Presented) The method of claim 16, further comprising generating on one or more computing devices one or more persistent components to maintain conversational state relating the associated method.

18. (Original) The method of claim 16, wherein the one or more declarative annotations further identify the associated method as being at least one of a start method, a continue method, and a finish method, wherein the start method applies to the start of a stateful conversation between a client and the web service, the continue method applies to the continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to the completion of an ongoing stateful conversation between a client and the web service.

19. (Original) The method of claim 16, wherein the one or more declarative annotations further identify the associated method as being a buffered method, wherein one or more queues are instantiated to temporarily store one or more requests for the identified method.

20. (Previously Presented) The method of claim 16, further comprising:  
generating on one or more computing devices a proxy object designed to facilitate interaction by the web service with an external web service.

21. (Previously Presented) The method of claim 20, further comprising:  
generating on one or more computing devices a unique identifier to identify a specific instance of the external web service.

22. (Original) The method of claim 16, wherein the source code is written in the Java programming language.

23. (Original) In a stateful web service, a method comprising:
- receiving a message requesting that a web service method be invoked;
  - parsing the message to identify the requested method;
  - determining whether the method is a stateful method based at least in part upon meta-data derived from one or more declarative annotations stored in association with object codes of the web service; and
  - dispatching the received request to invoke the identified stateful method.
24. (Original) The method of claim 23, wherein the message is received from a remote client.
25. (Original) The method of claim 23, wherein the message includes a globally unique conversational identifier identifying a specific conversational instance to facilitate management of multiple simultaneous conversations by the web service.
26. (Original) The method of claim 23, wherein the message is a SOAP based message.
27. (Original) The method of claim 26, wherein the conversational identifier is a GUID encapsulated in a header of the SOAP message.

28. (Original) The method of claim 23, wherein if a start method is invoked, a new instance of a conversation is created and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

29. (Original) The method of claim 23, wherein if a continue method or a finish method is invoked, a unique identifier provided by the client is identified and used to access a corresponding instance of a conversation.

30. (Original) The method of claim 29, wherein if a finish method is invoked, the corresponding instance of the conversation is destroyed by the web service logic after processing.

31. (Original) An article of manufacture comprising:  
a storage medium having stored therein a plurality of programming instructions,  
which when executed provide a graphical interface to facilitate specification  
of one or more declarative annotations within a procedural programming  
environment to modify an identified method of a stateful web service to cause  
a compiler to generate one or more persistent components to maintain  
conversational state related to the identified method.

32. (Original) The article of claim 31, wherein the one or more declarative annotations are specified within a source code representation of at least a portion of the web service based at least in part upon graphical input by a developer.

33. (Original) The article of claim 32, wherein the one or more declarative annotations are specified within a comment field preceding the identified method.

34. (Original) The article of claim 31, wherein the one or more declarative annotations are specified outside of a source code representation of at least a portion of the web service and associated with the identified method by the compiler.

35. (Original) The article of claim 31, wherein the one or more declarative annotations cause the compiler to generate a proxy object designed to facilitate interaction by the web service with one of an external web service or client.

36. (Original) The article of claim 35, wherein the one or more declarative annotations further cause the compiler to route asynchronous responses from the external web service to code specified by a developer of the web service.

37. (Original) The article of claim 35, wherein the one or more declarative annotations further cause the compiler to generate a unique identifier to identify a specific conversational instance of the external service.

38. (Original) An article of manufacture comprising:



a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to generate a stateful web service, which programming instructions when executed enable the apparatus to:

read a segment of procedural source code representing at least a portion of the web service;

parse the segment of source code to identify the presence of one or more declarative annotations identifying an associated method within the segment as being stateful;

generate one or more object codes defining one or more publicly accessible service components based at least in part upon the source code;

generate meta-data based at least in part upon the one or more declarative annotations;

associate the meta-data with the one or more object codes.

39. (Original) The article of claim 38, wherein the instructions when executed further enable the apparatus to generate one or more persistent components to maintain conversational state relating the associated method.

40. (Original) The article of claim 38, wherein the one or more declarative annotations further identify the associated method as being at least one of a start method, a continue method, and a finish method, wherein the start method applies to the start of a stateful conversation between a client and the web service, the continue method applies to the continuation of an ongoing stateful conversation between a client and the web service, and

the finish method applies to the completion of an ongoing stateful conversation between a client and the web service.

41. (Original) The article of claim 38, wherein the one or more declarative annotations further identify the associated method as being a buffered method, wherein one or more queues are instantiated to temporarily store one or more requests for the identified method.

42. (Original) The article of claim 38, wherein the instructions when executed further enable the apparatus to generate a proxy object designed to facilitate interaction by the web service with an external web service.

43. (Original) The article of claim 42, wherein the instructions when executed further enable the apparatus to generate a unique identifier to identify a specific instance of the external web service.

44. (Original) The article of claim 38, wherein the source code is written in the Java programming language.

45. (Original) An article of manufacture comprising:  
a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to generate a stateful web service, which programming instructions when executed enable the apparatus to

receive a message requesting that a method of the web service be invoked;

parse the message to identify the requested method;

determine whether the method is a stateful method based at least in part upon meta-data derived from one or more declarative annotations stored in association with object codes of the web service; and

dispatch the received request to invoke the identified stateful method.

46. (Original) The article of claim 45, wherein the message is received from a remote client.

47. (Original) The article of claim 45, wherein the message includes a globally unique conversational identifier identifying a specific conversational instance to facilitate management of multiple simultaneous conversations by the web service.

48. (Original) The article of claim 45, wherein the message is a SOAP based message.

49. (Original) The article of claim 48, wherein the conversational identifier is a GUID encapsulated in a header of the SOAP message.

50. (Original) The article of claim 45, wherein if a start method is invoked, a new instance of a conversation is created and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

51. (Original) The article of claim 45, wherein if a continue method or a finish method is invoked, a unique identifier provided by the client is identified and used to access a corresponding instance of a conversation.

52. (Original) The article of claim 51, wherein if a finish method is invoked, the corresponding instance of the conversation is destroyed by the web service logic after processing.